情 報 学 基 礎 14－7
（１９８９．７．２７）

テキストデータベース管理システム ＳＩＧＭＡとその利用

有川 節夫[1)] 篠原 武[2)] 宮原 哲浩[3)] 武谷 峻一[4)] 宮野 悟[1)]
竹田 正幸[5)] 大島 一彦[6)] 白石 修二[6)] 酒井 浩[7)] 山本 章博[8)]

1）九州大学理学部基礎情報学研究施設
2）九州工業大学情報工学部知能情報工学教室
3）九州大学教養部情報科学科
4）九州大学工学部中央計数施設
5）九州大学工学部電気工学科
6）福岡大学理学部応用数学教室
7）九州工業大学工学部電気工学科
8）九州大学総合理工学研究科情報システム学専攻

著者らが，九州大学大型計算機センターで1981年より公開しているテキストデータベース管理システムＳＩＧＭＡについて，その概要と基本的エンジンについて述べる．本システムによるフルテキストデータベースの構築例には，本センター公用データベース「トーマス・マン・ファイル」，「ゲーテ・ファイル」，「昆虫学データベース（ＫＯＮＣＨＵ）」などがあり，各分野の研究者に重宝されている．現在公開しているＳＩＧＭＡシステム第２版で基本的エンジンとして採用しているバタン照合アルゴリズムは，1バイト文字と２バイト文字が混在する日本語テキストをそのまま処理するので，日本語処理のためのオーバーヘッドは特になく，大規模なテキストを高速に検索することができる．文字を含むバタンも取り扱えるように基本的エンジンを改良する方法についても述べる．

A TEXT DATABASE MANAGEMENT SYSTEM SIGMA AND ITS USAGE

Setsuo ARIKAWA[1)], Takeshi SHINOHARA[2)], Tetsuhiro MIYAHARA[3)], Shun-ichi TAKEYA[4)], Satoru MIYANO[1)]
Masayuki TAKEDA[5)], Kazuhiko OSHIMA[6)], Shuji SHIRAISHI[6)], Hiroshi SAKAI[7)], Akihiro YAMAMOTO[8)]

1)Res. Inst. Fundamental Information Science, Kyushu Univ. 33, Fukuoka 812, JAPAN
2)Dept. Artificial Intelligence, Kyushu Institute of Technology, Iizuka 820
3)Coll. General Education, Kyushu Univ. 01, Fukuoka 810
4)Computation Center, Kyushu Univ. 36, Fukuoka 812
5)Dept. Electrical Engineering, Kyushu Univ. 36, Fukuoka 812
6)Dept. Applied Mathematics, Fukuoka Univ., Fukuoka 814-01
7)Dept. Electrical Engineering, Kyushu Institute of Technology, Kitakyushu 804
8)Dept. Information Systems, Kyushu Univ. 39, Kasuga 816

A general-purpose text database management system SIGMA was developed by the authors , and it is open to the public at Computer Center, Kyushu University since 1981. This paper describes the outline of the system and its main engine. Public Databases at the Computer Center "Thomas Mann File" , "Goethe File" and "A Database of Entomology KONCHU" are the examples of full text databases managemented by SIGMA, and are utilized by many researchers. Since the algorithms we developed process any Japanese texts which consist of both 1-byte and 2-byte characters with shift codes as they are, there is no additional overhead. The SIGMA system is used for the management of large scale full text database, the analysis of natural languages, the retrieval of bibliographic data, and so on. We also propose a matching algorithm for patterns containing pictures.

1. はじめに

パソコンやワープロの急速な発達に伴って，われわれの身のまわりには機械可読な文書が大量に作られ，テキストファイルとして蓄えられるようになっている．このようにして集積されるテキストファイルは，著者らが九州大学大型計算機センターで1981年より公開しているテキストデータベース管理システムSIGMA で管理するだけで，フルテキストデータベースとして運用することができる．つまり，ワープロを打つ感覚で手軽に大規模フルテキストデータベースを作成することができる．

本稿では，２章でSIGMA システムの概要について，３章でSIGMA システムの使用法の概略を述べる．４章では，九州大学大型計算機センター・公用データベース「トーマス・マン・ファイル」[11,13]，「ゲーテ・ファイル」[14,15]，「昆虫学データベース(ＫＯＮＣＨＵ)」[22]を中心にして，SIGMA によるフルテキストデータベース構築の実例を説明する．５章で基本的エンジンとして採用しているパタン照合アルゴリズムを説明し，文字種を含むパタンも取り扱えるように改良する方法を提案する．

2. SIGMA システムの概要

テキストデータベース管理システムSIGMA は，データを単純にテキストファイル，すなわち，一次元の文字列の形で管理し，ファイル全体を一度だけ先頭から一字ずつ走査するという一方向逐字処理に基本をおいている．フロッピーディスクなどに蓄えられているテキストファイルには，英語の文書，日本語のテキスト，DNA ファイル，作家の全集などがある．とくに,1バイト文字（半角の英字など）と２バイト文字（漢字など）から成る日本語テキストの増大が著しい．

SIGMA は，このようなテキストファイルの集積を管理し，フルテキストデータベースとして運用するシステムである．文字列という構造は，複雑な検索や，編集，検索されたレコードの再ファイル化を行うのに最も単純で基本的なデータ構造である．さらに，テキストファイルは，ワープロのユーザが作成する最も共通のデータでもある．SIGMA に取り込めるデータセットは，テキストファイルであれば，拡張子やレコード形式は何でもよい．また，フロッピーディスクのファイルも簡単に取り込むことができる．

図1(a)のテキストは，「トーマス・マン・ファイル」の一部である．もし“#@”を区切り語として選べば，１パラグラフが１レコードになる．区切り語の選択に応じて，レコードの概念も変化する．このような意味で，テキストファイルは，さまざま種類のレコードの総体を表現することができる．さらに，もし空白記号を区切り語として選べば，非常に細かい検索，つまり単語調査ができることになる．これは，テキストファイルを一方向逐字処理する検索システム，すなわちフルテキストデータベース検索システムの著しい特徴であり，他の情報検索システムには見られない性質である．著者らの開発した技法[5,21]は，一方向逐字処理を実時間可能なものとしている．例えば，SEARCHコマンドは１CPU 秒（応答時間で５秒）程度で２メガバイト（全角で百万字）のファイルを逐字検索することができる．しかも,1バイト文字と２バイト文字が混在する日本語テキストをそのまま処理するので，日本語処理のためのオーバーヘッドは特にない．

SIGMA システムは，テキストファイルに対する強力で高速な演算（複数文字列の同時検索，複数文字列の同時置き換え，自由な形式のファイルの並べかえなど）を行うことができる．この演算は，日本語テキストに対する高速な複数文字列照合アルゴリズムに基づく．このアルゴリズムを使えば，数万ものキーワードを領域的にも時間的にも効率的に検索することができる．

ファイルシステムは，単純ではあるが実用的な学習機能を実現可能にしている．システムとユーザとの応答は，常にファイル（LOG ファイルと呼ぶ）に記録されている．このLOG ファイルは，またテキストファイルであって，ユーザが与えたコマンドと入力の列から成る．同じコマンドを実行するプログラムとして，このファイルを直接実行することができる．

現在，SIGMA システムは，九大大型計算機センターのFACOM M-780 OSIV/F4 MSP 上で公開しており，多くの研究者から利用されている．また，個人用のデータベース以外に，グループで共有するためのデータベースを構築することもできる．さらに，くずかごの考え方によるファイルの保護機能などを持つ．

SIGMA は，きめの細かい検索，特に自然言語の解析などに威力を発揮している．例えば，和歌集を図1(b)のような形式でテキストファイル化しておけば，様々な検索を容易に行うことができる [9]．文献型データベースを構築するときには，図1(c)のような形式のテキストファ

```
#@
#0300907 DIE <GESCHICHTE <HANS <CASTORPS , DIE WIR
 ERZ=AHLEN WOLLEN , - NICHT UM SEINETWILLEN  ( DEN
N  DER <LESER WIRD EINEN EINFACHEN , WENN AUCH ANS
PRECHEN  DEN JUNGEN <MENSCHEN IN IHM KENNENLERNEN
 )  , SONDERN UM DER <GESCHICHTE WILLEN , DIE UNS I
N HÖHEM <GRADE ERZ=AHLENSWERT SCHEINT ( WOBEI ZU <
HANS <CASTORPS <GUNSTEN DENN DOCH ERINNERT WERDEN
SOLLTE , DA$ ES SEINE <GESCHICHTE IST , UND DA$ NI
CHT JEDEM JEDE <GESCHICHTE PASSIERT  >: DIESE <GE
SCHICHTE IST SEHR LANGE HER , SIE IST SOZUSAGEN SC
HON GANZ MIT HISTORISCHEM <EDELROST =UBERZOGEN UND
 UNBEDINGT IN DER <ZEITFORM DER TIEFSTEN <VERGANGE
NHEIT VORZUTRAGEN .
#@
#0300917 DAS W=ARE KEIN <NACHTEIL F=UR EINE <GESCH
```

図１(a).　「トーマス・マン・ファイル」

```
@
(T)ふるとしに春たちける日よめる$　　(A)在原元方$
#0001¥年の内に春はきにけり　ひととせをこぞとやいはん　ことしとやいはん$
@
(T)春たちける日よめる$　　(A)紀貫之$
#0002¥袖ひちてむすびし水のこほれるを　春立つけふの風やとくらん$
@
```

図１(b).　テキストファイル（和歌）

```
$
(TI) Efficient String Matching: An Aid to Bibliogr
aphic Search
(AU) Aho,A.V., Corasick,M.J.
(SO) Comm.ACM,Vol.18,pp.330-340,(1975)
$
(TI) テキストデータベース管理システムＳＩＧＭＡとその利用
(AU) 有川節夫ほか
(SO) 情報処理学会研究報告，89-FI-14,(1989)
$
```

図１(c).　テキストファイル（文献型）

イルを，エディタかワープロで作成して，SIGMA システムへ取り込むだけでよい．

したがって，SIGMA によるフルテキストデータベースの作成・維持は極めて容易であり，パソコン（と時間か予算）さえあれば，個人で大規模フルテキストデータベースを構築し，他の利用者に提供することも簡単にできる．しかも，検索は逐字的に行うため，転置ファイル方式の情報検索システムで用いられているストップワード（キーワードに指定できない語）を特に設定する必要はないので，検索の精度は高い．

3． SIGMA システムの使用法

SIGMA システムの使用法を，簡単に説明する．その詳細については，包括的なマニュアルである [9]を参照されたい（SIGMA システムの中で，"LIST S.'A71095B.MANUAL'" としてもよい）．

3．1 領域とファイル構成

SIGMA は，ファイルの高速処理および共有データベースの管理のため，独自のファイルシステムを有し，それによってファイルを管理している．また，オペレーティングシステムが管理しているテキストファイル（外部ファイルと呼ぶ）も簡単に参照できる．SIGMA が扱うファイルは，検索コマンドの索引付け用ファイルを除いてすべてテキストファイルである．

ファイルは用途に応じて数種類に分けられ，種類別にまとめて保管されている．同種類のファイルの集まりを領域と呼ぶ．領域には，個人用のファイルを置くMEMO領域，データベースを共有するための SIGMA領域，外部ファイルの集まりである外部領域がある．各領域間でファイルを転送することができる．

MEMO領域は，図２に示すように，５個の部分領域（区域と呼ぶ）から構成され，各区域間でファイルを転送することができる．メモ区域には，個人用の名前つきのファイル（メモファイルと呼ぶ）が置かれる．他に，いろいろな作業の中間結果を置くための作業用区域，システムとの交信記録（ログファイルという）を置くためのログ区域，検索コマンドの索引付け用ファイルを置くための索引用区域，MEMO領域のファイルのバックアップを置くためのバックアップ区域がある．

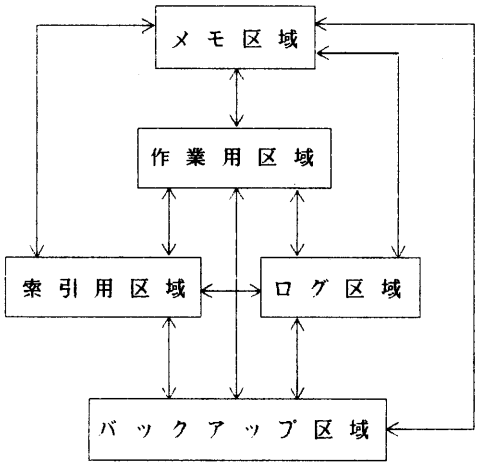

図２． MEMO領域のファイル構成

SIGMA では，ふだんは作業用区域のファイルを用いて，検索・置き換え・ソートなどを行う．個人用に整理したければ，名前をつけてメモファイルとすればよい．メモ区域を除く全ての区域は，底のあるスタックとして構成され，常に最新のファイルがそのトップに置かれ，除去される場合はそのスタックの底にある最古のファイルがバックアップ区域のトップに移される．こうして丁度屑籠のように，容量に余裕がある限り，除去されたファイルも保存される．これによって，操作ミスによるデータの損失を防いでいる．

3．2 コマンド一覧

SIGMA では表１のコマンドが使用可能である．コマンドを入力する際には，下線部まで入力すれば十分である．

次に，SIGMA の主要な３つのコマンドSEARCH,SORT,REPLACEについて説明する．

3．3 検索コマンド（SEARCH）

SEARCHコマンドの第一の特長は，区切り語と，キーワードの集合からパタン照合機械（pmm） と呼ばれる一種

表１． SIGMA のコマンド一覧

CATENATE	:ファイルを連接する．
COMMENT	:ファイルにコメントをつける．
COPY	:ファイルの複製を作る．
DDTRECTORY	:ディレクトリ（ファイル名一覧）を詳細情報とともに表示する．
DELETE	:ファイルを除去して（消去ではない），バッファ区域のトップに置く．
DIRECTORY	:ディレクトリを表示する．
ECHO	:ログファイルをコマンドプロシジャとして実行中に，実行内容を表示するかどうかを設定したり，端末にメッセージを出力したりする．
END	:SIGMAシステムを終了する．
GET	:指定するファイルを作業用区域のトップに置く．
HELP	:SIGMAで使用できるコマンドや用語の説明を表示する．
HEXDUMP	:ファイルの16進ダンプをとる．
KEYIN	:作業用区域のトップにあるファイルに，キーボードから直接入力する．
LIST	:ファイルの内容を表示する．
LOAD	:指定するファイルの内容を，作業用区域のトップにあるファイルに読み込む．
LOOK	:作業用区域のトップにあるファイルの内容を表示する．
MOVE	:ファイルを移動する．
NICKNAME	:ファイルに別名をつける．
PASSNUMBER	:パスナンバーを設定，変更する．
PROFILE	:システム定数を表示したり設定したりする．
PUT	:作業用区域のトップにあるファイルを指定する所に置く．
REFILE	:検索結果を再ファイル化する．
REPLACE	:複数文字列の同時置き換えをする．
SAVE	:作業用区域のトップにあるファイルの内容を指定するファイルへ書き出す．
SEARCH	:複数キーの同時逐字検索を行う．
SETMEMO	:MEMO領域を初期化する．
SETSIGMA	:SIGMA領域を初期化する．
SORT	:レコードのソーティングをする．
SPACE	:MEMO領域の使用可能量を表示する．
TSS	:TSSコマンドを呼び出す．

の有限オートマトンを作り，それが対象となるファイル，すなわちテキストファイルの上を先頭から末尾までただ一度走査する間にすべての作業を行う点にある．この他の主な特長は次の諸点にある．

(1) 自由なレコード形式に対して適用できる．テキストファイルは単なる文字列であるが，検索の単位となるレコードはレコード区切り語と呼ばれる文字列にはさまれた部分列として，仮想的に設定する．このため，テキストファイルの形式や検索の目的に応じて，自由にレコードを設定することができる．たとえば，段落や文章などは特定の文字列（復改記号や句点）で区切られているので，検索の単位を段落とするか文章とするかに応じて，それらをレコード区切り語に指定すればよい．また，検索時の意味の単位（論理式を評価する単位）も，項目区切り語と呼ばれる文字列を指定することによって設定できる．これを使えば，レコードを項目に区切ってきめ細かい検索条件を設定することができる．

(2) X1...X2...X3の形のキーワードの指定ができる．トリプル・ドット "..." は任意の文字列（空でもよい）を表わす．これによって，キーワードの出現順序に意味をもたせた検索が可能になる．

(3) 複数質問を同時に処理できる．SEARCHコマンドは一度ではあるが，ファイル全体を走査するので，処理に要する時間は必然的に対象ファイルの大きさに比例したものとなる．しかし，最大99個のキーワードを組み合わせて，論理式を99個，結果を得るための質問式（＝論理式）を32個同時に処理できるので，複数質問を同時に処理すれば，質問１個あたりの処理時間は比較的に短いことになる．もちろん，SEARCHコマンドで用いるパタン照合機械は非常に高速であるため，数メガバイト程度のテキストならば実用的時間内に処理できる．

(4) 検索能力が高い．キーワードの出現を単にマークするのではなく，出現回数をカウントするようにし，さらに論理式中ではキーワードの出現回数を用いた演算を可能とした．これにより，通常の論理演算では表わすことができないような質問も処理できる．

３．４　ソートコマンド（SORT）

SORTコマンドは，SEARCHコマンドと同様に，レコード区切り語や切り出しパタン（切り出すべきキーを指定するもの）からパタン照合機械を作り，それが対象とするテキストの上を先頭から末尾までただ一度だけ走る間に必要なキーを切り出して，ソートするものである．

通常の，例えば表形式のデータに対するソートでは，単に第１キー，第２キー，…となる欄を指定すればソートできるが，このSORTコマンドは，テキストファイルを対象としているため，さらに以下に述べるいろいろな特長や機能を備えている．

(1) SEARCHコマンドの項で述べたように，自由なレコード形式に対して適用できる．

(2) 通常のソートと同様に，優先度がついた複数のキーを指定できる．

(3) １つのキーの切り出し指定（切り出し仕様と呼ぶ）では，仮想レコード中のどの範囲からキーを切り出すかを指定でき（範囲記述部により指定），またどのようなキーを切り出すかを，文字種のパタンで指定できる（読み込み記述部により指定）．キーの切り出しは，指定された範囲に入った直後ではなく，その後で指定された切り出しパタンの最初の文字種が見つかった時点で開始し，以下切り出しパタン全部，または切り出しパタンと異なる文字が見つかるまで行なわれる．これによって，例えば切り出しパタンとして漢字５文字を指定して氏名を切り出そうとした場合，最長５文字までの氏名を切り出すことになり，３文字や４文字など５文字に満たない氏名も切り出すことができる．また，切り出しパタンとして，複数の選択的なパタンも指定でき，この場合は最初に先頭の文字種が見つかったパタンで切り出される．

(4) 切り出し仕様では，各種のオプションも指定できる．これには，昇順／降順，キーの左寄せ／右寄せ，左／右からのソート，キーが欠落した場合の扱いなどがある．

なお，通常のソートでは，各キーの切り出し仕様に基づいた切り出しが１つの仮想レコードで１回だけ行われる．単語単位のソートを行おうとする場合には，英文などのように単語が空白で区切られているときは，空白を区切り語と指定して１単語を１レコードとすれば，このままでよい．しかし，日本語文のように単語が区切られていない場合には難しいので，切り出し範囲の中で指定された切り出しパタンの文字列を，可能な限り何回でも切り出して，その１つ１つをキーとするモードも付加した．

(5) ソートされた結果をどのように再ファイル化するかに関しても，豊富な組合せを用意した．

３．５　置き換えコマンド（REPLACE）

ファイルの中の指定された文字列をことごとく指定された他の文字列で置き換えることができる．これは通常のエディタの置き換えコマンドを一般化したもので，次のように入力する．

```
DO:REPLACE
XO1:=x1
YO1:=y1
XO2:=x2
YO2:=y2
.....
Xn:=xn
Yn:=yn
Xn+1:=↙
INPUT-FILE:=filename1
OUTPUT-FILE:=filename2
```

これによって，filename1という名前のファイルの中に含まれている文字列xiをことごとく文字列yiで置き換えたファイルが filename2に作られる．なお，使用にあたって以下のことに注意されたい．

(1) yiは空列（すなわち，Yi:=↙）でもよいが，xiには空列は許されない．（↙は復改のみの入力を表す．）

(2) ある文字列xiが検出され，これがyiで置き換えられた時点で，文字列x1,x2,･･･,xnを探す作業は初期化され，このxiの次の文字から新しい置き換え作業が繰り返される．

(3) 文字列 xi,yiの登録において，xi＝xj（i＜j）となった場合には，始めの方の

```
Xi:= xi
Yi:= yi
```

は無視される．したがって，これを置き換えの指定の訂正に使うことができる．すなわち，第i番目の指定を訂正するには，

```
Xj:= xi
Yj:= yj
```

のようにXjに訂正したい文字列xiを，またYjに正しく置

き換えるべき文字列yjを割り当てればよい．
(4) REPLACE コマンドは左から右への一方向の逐字処理により置き換えを行う．キーワードに指定した文字列が重複している場合は，左から見てなるべく長い文字列を置き換え，バックトラック（後戻り）は行わない [6]．下図で説明しよう．この例では，左のx2はより長いx1に含まれているので無視される．左のx4は，x3より右にあるので無視される．これは，バックトラックを行わないためである．置き換えられるのは，x1,x3,x5,右のx2,右のx4である．

```
                X1      X3    X5      X4
置き換え前      ああああいいいううえおおおあいうえお
                  X2        X4      X2
```

```
置き換えの指定  XO1:=ああいい
                YO1:=アアイイ
                XO2:=あい
                YO2:=ＡＩ
                XO3:=うう
                YO3:=ウウ
                XO4:=うえお
                YO4:=ＵＥＯ
                XO5:=おお
                YO5:=オオ
                XO6:=↙
置き換え後      あアアイイいウウえオオＡＩＵＥＯ
```

３．６ LOG ファイルの利用法

SIGMA システムでは，利用者との交信記録（ LOG）がMEMO領域の LOG区域に自動的に作られる．こうしてできたファイルを LOGファイルという．LOG ファイルは，他のファイルと同様に文字列の形をしたもので，内容はシステムが端末に出すプロンプトとそれに対する利用者の応答の記録である．LOG ファイルを，そのままあるいは変更を加えて一種のコマンドプロシジャとして利用すれば非常に便利である．

SIGMA システムには SIGMA状態とDO状態があり，プロンプトはそれぞれ“SIGMA>”，“DO:” となっている．SIGMA システムが呼び出されると，まず SIGMA状態になり，END 以外のコマンドを投入すると， LOGがとられ始める．SIGMA 状態でのコマンドの処理が終わると，DO状態になる．DO状態で ENDコマンドを投入すると，ここまでが１回の交信として LOG区域のトップに置かれて “L” という名前のファイルになり，システムはSIGMA 状態になる．ここで ENDコマンドを投入すればSIGMA システムが終了するし，他のコマンドを投入すれば次の交信がとられる．

最新のLOG ファイルの内容を実行するには，“.L” と打てばよい（下記実行例参照）．SEARCHやSORTコマンドではキーワードや論理式，キーの切り出しなどをきめ細かに指定できるが，定型的なテキスト処理を行う場合それらを毎回キーボードから入力していたのでは効率が悪い．テキスト処理を行うLOG ファイルに名前をつけておいて，コマンドプロシジャとして実行し，処理をするファイル名だけを入力するようにすれば便利である．

```
READY
SIGMA
SIGMA>LIST
FILE:=W.2
This is a text.
DO:END
SIGMA>LIST L
DO:LIST
FILE:=W.2      ┐
               ┘ Lの内容
DO:END
DO:.L
DO:LIST
FILE:=W.2
This is a text.
DO:END
DO:MOVE L LOOK2
DO:END
SIGMA>END
```

４．フルテキストデータベース

九大大型計算機センター・公用データベース「トーマス・マン・ファイル」[11,13]，「ゲーテ・ファイル」[14,15]，「昆虫学データベース（ＫＯＮＣＨＵ）」[22]を中心にして，SIGMA によるフルテキストデータベース構築の実例を説明する．データベースの詳細および具体的な利用法については，それぞれ文献および図４の実行例を参照されたい．

４．１ トーマス・マン・ファイル

九大言語文化部樋口忠治教授が，S.Fischer書店刊Thomas Mann全集（約１万１千ページ，約40万行）を入力して作成したデータベースである（約26メガバイト）．1983年に全集の半分程度で公開が始まり，1986年に全集すべてのテキストファイル化が終了し公開された．全集すべての逐字検索に13 CPU秒程度しか要しないので，従来の言語科学ではまったく手がつけられなかった研究分野が生まれ，語の用法に関する新しい知見が得られるようになった[12]．文学・語学におけるテキストデータベース利用の先駆けであり，研究方法の革新をもたらすものである．

原則として，１作品を１ファイルとしてSIGMA 領域のファイルとして管理されている．# は文章の，#@はパラグラフの区切りをレコード区切り語である．ファイルは，

… #VVPPPLL 文章　#VVPPPLL 文章　#VVPPPLLL …

という形式をしており，VVは巻，PPP はページ，LLは行を表した数字である．

４．２ ゲーテ・ファイル

樋口教授が，J.W.v.Goetheのハンブルク版全集（全14巻）をテキストデータベース化したものである（約15メガバイト）．ファイルは，トーマス・マン・ファイルと同様な形式で作成され，管理されている．

４．３ 昆虫学データベース（ＫＯＮＣＨＵ）

九大農学部昆虫学教室多田内修助手が，1900年以降の国内の主要昆虫学雑誌について，各文献中に記載されている種や属といった分類単位を一つのレコードとして，構築したものである（現在，約１万２千件）．単なる文献データベースではなく，昆虫の様々な属性をデータとして含む新しいファクトデータベースとしての性格を持っている．しかも，フルテキストデータベースとして作成されているため，精度の高い逐字検索ができる．順次対象雑誌を広げ，データベース化していく予定である．

ファイルに含まれる情報とタグは次のようになっている．分類単位名（学名）(TAX)，分類単位名（和名）(JTAX)，著者名(AU)，論文タイトル(T)，雑誌名(J),巻・号・頁(VNP)，刊行年(Y)，目名(OR)，科名(FAM),同物異名

(SYN)，分布(DST)，キーワード(KEY)．

また，国内に産する昆虫すべて（約３万種）について学名・和名・分布から成る目録を作成中であり[19,28]，そのテキストデータベース化も予定している．

５．パタン照合アルゴリズム

SIGMA システムで，テキスト処理の基本的エンジンとして採用しているパタン照合アルゴリズムについて，その概要を述べ，文字種を含むパタンも扱えるように改良する方法を提案する．

５．１　一方向逐字処理

テキストからキーを検出する手法をパタン照合と呼ぶ[7]．その手法の代表的なものとして，Knuth-Morris-Pratt[17]やBoyer-Moore[10]，Aho-Corasick[1]のものなどが知られている．前の２つは１度に１個のキーを検出するものであるが，Aho-Corasickの手法は同時に複数のキーを検出するものである．SIGMA では，この Aho-Corasick のパタン照合機械を処理の基本に置いた．それは単純な一次元文字列という構造の上で高度な処理が可能だからである．

Aho-Corasick のパタン照合機械とは，具体的には与えられたキーの集合からそれらを検索する一種の有限オートマトンを構成し，このオートマトンで入力テキストを先頭から１回だけ走査する一方向逐字処理により，キーの存在をチェックするものである．従って，検索には入力テキストの長さに比例した時間が必要なのはやむをえない．

しかし，この手法を SIGMA システムで実現するにあたっては，オートマトンの遷移表のサイズの最小化や，文字コード分割（英字１文字を２つに分割）と表一瞥[18]による高速化などを行い，アルゴリズムを大幅に改良している [5]．また，SIGMA ではこのパタン照合機械を，複数キーの同時検索だけでなく，ソーティング用キーの切り出し，複数文字列の同時置き換えなど多くの場面で活用している．

５．２　日本語テキスト用パタン照合アルゴリズム

現在公開しているSIGMA システム第２版で，テキスト処理の基本的エンジンとして採用している日本語テキスト対応パタン照合アルゴリズムについて述べる．

日本語テキストは英文テキストと異なり字種が多く，そのため計算機内部では通常１文字が２バイト（英字２字分）で表現されている．また，多くの場合１文字１バイトの文字との混在も許されている．こうした字種が多く異なる符号長の文字が混在するという状況を考慮して，著者らは新しく日本語テキストのためのパタン照合アルゴリズムを開発した[21]．

このアルゴリズムは，先に開発した英文テキスト用の高速アルゴリズム [5]で用いた，英字１文字を２つに分割する方法を拡張することで実現できた．具体的には，２バイト系文字と１バイト系文字の切り替え用のシフトコードに関する遷移を，オートマトンの構成時に遷移表の核として初期設定しておくことである．構成の概要を図３に示すが，図中の２桁の文字はシフトコードを構成する文字で，16進数である．図中のコード系は，JIS C 6226に従った．また，遷移は１バイト単位で行うことになっているが，Mは２バイト系文字が遷移する場合の中間状態に対応する．実際の第２版の開発は，FACOM OS IV 上の JEFコードに従い，高速化のため４ビット単位で遷移する手法によった．

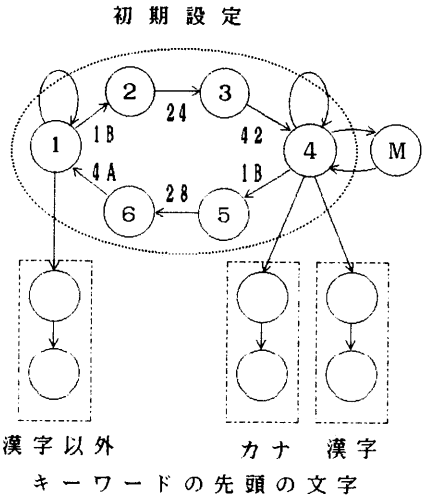

図３．日本語テキストのためのパタン照合機械の構成

実現したアルゴリズムは，１バイト文字と２バイト文字が混在する日本語データを，文字分割の方法を用いてそのまま４ビット単位で照合するものであり，すべての文字を２バイト化するなどの前処理（正規化，[29]pp.124-137）を必要としない．つまり，日本語処理のためのオーバーヘッドは特にないといってよい．よって，実行速度は英文テキスト用のパタン照合アルゴリズムと変わらない．実際，第１版と同じくFACOM M-382 上でも，第２版のSEARCHコマンドは，英文テキスト専用の第１版SEARCHコマンドとほぼ同じ速度で，日本語テキストを処理することを確かめた．

５．３　文字種を含むパタンを対象とする照合アルゴリズム

より柔軟できめの細かいテキスト処理を実現するために，SIGMA システムで基本的エンジンとして採用しているパタン照合アルゴリズムを改良する方法を述べる．通常のAho-Corasick型パタン照合アルゴリズムは，単純な文字列だけを対象とするものであった．ここでは，文字種（ピクチャという）を含むパタンを取り扱うことを考える．

例えば，英小文字(a,b,...,z) を表すピクチャをA とし，数字(0,1,...,9) を表すピクチャをN とするとき，aA,a7NNNNaなどのパタンをテキスト中より検出することを考える．最も単純には，このようなパタンに対して，パタンに属するすべての文字列から文字列照合機械を構成する方法が考えられる．すなわち，aAに対しては，文字列aa,ab,...,azから照合機械を構成する．しかし，この方法は状態数の爆発を引き起こす．

Takeda[24,25] は，文字とピクチャの混在するパタン

を同時に複数個探索するためのパタン照合機械の効率的な構成法を提案している．この方法で構成される照合機械の状態数は，必ずしも最小ではないが，多くの場合それに近いものになっている．そして，構成された照合機械は，通常のAho-Corasick型文字列照合機械と同様に，テキスト長の線形時間で走ることができる．

１バイト文字を２つの部分コードに分割する方法は，残念ながら使うことはできない．しかし，“ひらがな”や“漢字”のような２バイトの文字種は，それらを２つのピクチャの連接とみなすことにより，取り扱うことができる．

この方法は，SEARCHコマンドだけでなくREPLACE コマンドに対しても適用することができる [26,27]．例えば，1967,1979,1985,...をそれぞれ'67,79,'85,...へ置き換えたい場合を考えよう．REPLACE コマンドを用いると，置き換えのすべての対(1967,'67),(1979,'79),(1985,'85),...を与えなければならず，構成される照合機械の状態数もそれに応じて増大する．これに対し，ピクチャの考え方を使えば，19NN → 'NN のような置き換えの形式を与えるだけでよく，照合機械はパタン19NNに対して構成されるので状態数を小さく抑えることができる．現システムより強力なテキスト処理を実現するために，ここで述べたアルゴリズムを用いて基本的エンジンを改良することを計画している．

６．　おわりに

一方向逐字処理に基づいたテキストデータベース管理システムSIGMA について，システムの概要とフルテキストデータベース構築の実例を説明した．

ここで，SIGMA システムの開発の歴史に簡単にふれる．

０版：	1979年	ミニコン版プロトタイプ完成（九大理学部基礎情報学研究施設 PANAFACOM U-400）
１版：	1981年	SIGMA 第１版公開（九大大型計算機センター FACOM M-200）
	1983年	高速パタン照合アルゴリズムの開発
	1985年	日本語テキスト用のパタン照合アルゴリズムの開発
２版：	1987年	日本語テキスト対応 SIGMA第２版公開（九大大型計算機センター FACOM M-780）

SIGMA によるフルテキストデータベース運用に関心がある方には，九大大型計算機センター公用データベース制度[20]を利用しての構築，大学間ネットワークを通じての利用をすすめたい．

第２版の開発に参加していただいた川崎洋治氏，井上仁氏，湯浅寛子氏ならびに原口誠助教授に深謝いたします．SIGMA システムに関してユーザの立場から有益な助言をいただいた九大言語文化部樋口忠治教授ならびに九大農学部多田内修助手に感謝します．

［参考文献］

[1] Aho,A.V. and Corasick,M.J.: Efficient String Matching : An Aid to Bibliographic Search, Comm. ACM, Vol.18, pp.333-340 (1975).
[2] Arikawa, S. and Kitagawa, T.: Multistage Information Retrieval System Based on Researcher Files, Proc. 2nd USA-Japan Computer Conf., pp.149-153 (1975).
[3] 有川，篠原，白石，玉越：研究者向き情報システムSIGMA について，九州大学大型計算機センター広報，Vol.14, No.4, pp.550-573 (1981).
[4] Arikawa,S., Shinohara,T., Shiraishi,S. and Tamakoshi,Y.: SIGMA - An Information System for Researchers Use, Bull. Inform. Cybernetics, Vol.20, No.1-2, pp.97-114 (1982).
[5] Arikawa,S. and Shinohara,T.: A Run-Time Efficient Realization of Aho-Corasick Pattern Matching Machines, New Generation Computing, Vol.2, No.2, pp.171-186 (1984).
[6] Arikawa,S. and Shiraishi,S.:Pattern Matching Machine for Replacing Several Character Strings, Bull. Inform. Cybernetics, Vol.21, No.1-2, pp.101-111 (1984).
[7] 有川，篠原：文字列パターン照合アルゴリズム，コンピュータソフトウェア,Vol.4,No.2,pp.2-23(1987).
[8] 有川ほか：テキストデータベース管理システムSIGMA について，情報処理学会研究報告，Vol.87, No.65 (FI-6), pp.6.4.1-6.4.8 (1987).
[9] 有川ほか：テキストデータベース管理システムSIGMA 第２版について，九州大学大型計算機センター広報，Vol.20, No.6, pp.517-581 (1987).
[10] Boyer,R.S. and Moore,J.S.: A Fast String Searching Algorithm, Comm. ACM, Vol.20, pp.762-772 (1977).
[11] 樋口，篠原:公用データベース トーマス・マン・ファイル／SIGMAの公開，九州大学大型計算機センター広報 Vol.16, No.4, pp.379-393,(1983).
[12] 樋口忠治：「前置詞」bis の用法について，独仏文学研究，Vol.36, pp.183-210 (1986).
[13] 樋口，篠原：テキストデータベース「トーマス・マン・ファイル」の完成と再編成について，九州大学大型計算機センター広報，Vol.20, No.6, pp.582-596 (1987).
[14] 樋口忠治：テキストデータベース「ゲーテファイル」の公開，九州大学大型計算機センター広報,Vol.21, No.3,4, pp.167-176 (1988).
[15] 樋口忠治：ゲーテファイルの完成について，九州大学大型計算機センター広報，Vol.22, No.1, pp.1-5 (1989).
[16] Inose, H. (ed.): Scientific Information Systems in Japan, North-Holland (1981).
[17] Knuth,D.E., Morris, J.E. and Pratt,V.R.:Fast Pattern Matching in Strings, TR CS-74-440, StanfordUniv. (1974).
[18] Knuth,D.E.: The Art of Computer Programming, Vol.3, Sorting and Searching, Addison Wesley,1973.
[19] 九州大学農学部昆虫学教室，日本野生生物研究センター（編），日本産昆虫総目録（1989）．（印刷中）
[20] 松尾，村上，高木：公用データベースについて，九州大学大型計算機センター広報，Vol.15, No.2, pp.222-227 (1982).
[21] 篠原，有川：日本語テキスト用の Aho-Corasick型パターン照合アルゴリズム，情報処理学会研究報告，Vol.86, No.48 (NL-52), pp.52.4.1-52.4.8 (1985).
[22] 多田内修：SIGMAによる公用データベース(KONCHU)の公開とその利用法，九州大学大型計算機センター広報，Vol.20, No.6, pp.597-614 (1987).
[23] M.Takeda, A Proof of the Correctness of Uratani's String Searching Algorithm, RIFIS-TR-CS-7, Research Institute of Fundamental Information Science, Kyushu Univ., Oct. 1988.
[24] M.Takeda, A Fast Matching Alogrithm for Patterns with Pictures, RIFIS-TR-CS-11, Research Institute of Fundamental Information Science, Kyushu Univ., Mar. 1989.
[25] 竹田正幸：固定文字列と文字種の混在するパターンを対象としたAho-Corasick型パターン照合機械の構成法，九州大学大型計算機センター，計算機科学研究報告第6号，pp.29-51 (1989)
[26] M.Takeda,Efficient Multiple String Replacing with Pictures, RIFIS-TR-CS-13, Research Institute of Fundamental Information Science, Kyushu Univ., Mar. 1989.
[27] 竹田正幸：ピクチャ・パターン照合アルゴリズム，京都大学数理解析研究所集会「計算アルゴリズムと計算量の基礎理論」講究録（1989）．（掲載予定）
[28] M.Takeda, T.Miyahara: Query Processing for Structured Text as a Relation, RIFIS-TR-CS-15, Research Institute of Fundamental Information Science, Kyushu Univ., Mar. 1989.
[29] 生島和夫編：日本語処理のためのプログラミング支援環境の構築，九州大学工学部情報工学科 1986.

```
READY
SIGMA
SIGMA>DIR S.'A70152B.MANN.*'
PASSNUMBER:=R5X8K&W%M%B#↙
MANN.BB
MANN.BEISPIEL
MANN.DF
MANN.ERZ
MANN.EW
MANN.FK
MANN.JS1
MANN.JS2
MANN.KH
MANN.LT
MANN.NT
MANN.RA1
MANN.RA2
MANN.RA3
MANN.RA4
MANN.ZB

        TOTAL =  16          PREFIX = A79999A
 3321 SECTOR(S) AVAILABLE
DO:LIST S.'A70152B.MANN.JS1'
#@
#0400909 TIEF IST DER <BRUNNEN DER <VERGANGENHEIT
.
#0400909 SOLLTE MAN IHN NICHT UNERGR=UNDLICH NENNE
N ?
#@
#0400911 DIES N=AMLICH DANN SOGAR UND VIELLEICHT E
BEN DANN , WENN NUR UND ALLEIN DAS <MENSCHENWESEN
ES IST , DESSEN <VERGANGENHEIT IN <REDE UND <FRAGE
 STEHT : DIES <R=ATSELWESEN , DAS UNSER EIGENES NA
T=URLICH-LUSTHAFTES UND =UBERNAT=URLICH-ELENDES <D
ASEIN IN SICH SCHLIE$T UND DESSEN <GEHEIMNIS SEHR
BEGREIFLICHERWEISE DAS <A UND DAS <O ALL UNSERES <
REDENS UND <FRAGENS BILDET , ALLEM <REDEN <BEDR=AN
GTHEIT UND <FEUER , ALLEM <FRAGEN SEINE <INST=ANDI
GKEIT VERLEIHT .
#0400918 DA DENN NUN GERADE GESCHIEHT ES , DA$ , J
.............
!
DO:SEARCH
     RECORD DELIMITERS
D01:=#
D02:=↙
     ITEM DELIMITERS
D02:=↙
     KEYWORDS
A01:= WENN ... AUCH
A02:= WIE ... AUCH
A03:=↙
     LOGICAL FORMULAS
Z01:=A1
Z02:=A2
Z03:=A1.A2
Z04:=A1,A2
Z05:=↙
REPORT(Y/N)?↙
FILE:=S.'A70152B.MANN.JS1'
    RETRIEVED TEXTS
QUESTION 01 (Z01) =         82         82
QUESTION 02 (Z02) =        144        144
QUESTION 03 (Z03) =         15         15
QUESTION 04 (Z04) =        211        211
TOTAL             =        211        211
CPU (SEC/1000)    =        854        854
FILE:=↙
DO:REFILE
REPORT(Y/N)?N
QUESTION:=3
NEW RECORD DELIMITER:=#
NUMBERING(N/Y)?↙
OUTPUT-FILE:=W
QUESTION:=↙
DO:LOOK
#0405932 IN DER <UMGEBUNG GAB ES DER <B=AUME MEHR
, WENN AUCH SO EHRW=URDIGE , WIE DER GESONDERT STE
HENDE , SONST NICHT :  VON DERSELBEN <GATTUNG SOWO
HL , WIE AUCH GRO$BELAUBTE <FEIGENB=AUME UND <STEI
NEICHEN , DIE AUS IHREN <ST=AMMEN <LUFTWURZELN IN
DEN ZERTRETENEN <GRUND ENTSANDTEN UND DEREN BEST=A
NDIGES , VOM <MONDE GEBLEICHTES <GR=UN . ZWISCHEN
<NADEL UND <LAUB DIE <MITTE HALTEND , DORNIGE <F=A
CHER BILDETE .
#0413501 WENN ER SICH SP=ATER GANZ ZU DEN <LEUTEN
.............
!
DO:DIR S.'A71414B.*'
PASSNUMBER:=R5X8K&W%M%B#↙
ESAKIA
MUSHI

        TOTAL =   2          PREFIX = A79999A
 2624 SECTOR(S) AVAILABLE
DO:SEARCH
     RECORD DELIMITERS
D01:=#
D02:=↙
     ITEM DELIMITERS
D02:=!
D03:=↙
     KEYWORDS
A01:=(TAX) Palaeorhiza
A02:=(DST)...New Guinea
A03:=(DST)...Australia
A04:=Yakushima
A05:=Amami-Oshima
A06:=シェーファー
A07:=↙
     LOGICAL FORMULAS
Z01:=A1
Z02:=A2
Z03:=A3
Z04:=A4
Z05:=A5
Z06:=A6
Z07:=Z1.Z2
Z08:=Z1.Z3
Z09:=Z4.Z5
Z10:=↙
REPORT(Y/N)?↙
FILE:=S.'A71414B.ESAKIA'
    RETRIEVED TEXTS
QUESTION 01 (Z01) =        150        150
QUESTION 02 (Z02) =        399        399
QUESTION 03 (Z03) =         54         54
QUESTION 04 (Z04) =        117        117
QUESTION 05 (Z05) =        145        145
QUESTION 06 (Z06) =          1          1
QUESTION 07 (Z07) =        125        125
QUESTION 08 (Z08) =         10         10
QUESTION 09 (Z09) =         42         42
TOTAL             =        673        673
CPU (SEC/1000)    =        923        923
FILE:=↙
DO:REFILE
REPORT(Y/N)?N
QUESTION:=6
NEW RECORD DELIMITER:=#
NUMBERING(N/Y)?↙
OUTPUT-FILE:=T
QUESTION 06 (Z06) =          1
#
(TAX) Eurytoma schaeferi Yasumatsu et Kamijo, n. s
p.
(JTAX)シェーファーカタビロコバチ
(AU)  YASUMATSU, Keizo;KAMIJO, Kazuaki
(T)   Chalcidoid parasites of Dryocosmus kuriphilu
s Yasumatsu (Cynipidae) in Japan, with description
s of five new species (Hymenoptera)
(J)   Esakia
(VNP) (14): 106-109
(Y)   1979
(OR)  Hymenoptera
(FAM) Chalcidoidea
(DST) Japan(Hokkaido, Honshu, Kyushu, Satsunan Is.
)
(KEY) systematics;new species;description(female,
male);type(female:Kyushu Univ., No. 2144);type loc
ality(Teineyama, Hokkaido);biology;parasite of Dry
ocosmus kuriphilus;chestnut
#
QUESTION:=↙
DO:END
SIGMA>END
READY
```

図4．　実行例